Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (currently amended) An interprocessor communication (IPC) network, comprising:

an IPC stack including:

a presentation manager;

a session manager coupled to the presentation manager; and

a device interface coupled to the session manager;

a port coupled to the IPC stack;

a component coupled to the IPC stack; and whereby the session manager can dedicate use of the port to the component;

wherein the session manager dedicates the port for use by the component for a particular service.

2. (original) An IPC network as defined in claim 1, wherein the component comprises a software thread.

3. (original) An IPC network as defined in claim 1, wherein the session manager dedicates the port in response to receiving a request from the component.

4. (canceled)

5. (currently amended) An IPC network as defined in claim [[4]] 1, wherein the session manager dedicates the port by linking an opcode corresponding to the particular service to the port.

6. (original) An IPC network as defined in claim 5, wherein any IPC message sent by the component or other components coupled to the IPC stack, that carries the opcode linked to the port is routed through the port.

7. (original) An IPC network as defined in claim 1, wherein the session manager adds a command header to data sent by the component, wherein the command header causes the port to perform a certain co-processing task to the data prior to the data being sent from the port.

8. (original) An IPC network as defined in claim 7, wherein the co-processing task performed by the port provides for one of rate conversion and summing two or more data streams together.

9. (original) An IPC network as defined in claim 7, wherein the session manager forwards a request to the device interface whenever it wants to dedicate the port to a particular type of service.

10. (currently amended) An IPC network as defined in claim [[4]] 1, wherein once the port is dedicated, the component can communicate with a second component via

the port without having to add IPC headers to the communications between the component and the second component.

11. (original) A method for providing a port in an interprocessor network having at least one component coupled to an Interprocessor Communications (IPC) stack that includes a session manager, the method comprising the steps of:

dedicating the port to a particular type of service by the session manager; and

routing messages sent by the at least one component using that service to the port.

12. (original) A method as defined in claim 11, wherein the step of dedicating the port comprises linking at least one opcode that corresponds to the service to the port.

13. (original) A method as defined in claim 12, wherein the session manager dedicates the port in response to receiving a request from one of the at least one components.

14. (original) A method as defined in claim 12, wherein the session manager initiates dedication of the port.

15. (original) A method as defined in claim 12, further comprising the step of:

routing any messages that carry the at least one opcode associated with the port sent by the at least one component through the port.

16. (original) A method as defined in claim 11, wherein the session manager forwards a request to a device interface that is part of the IPC stack whenever it wants to dedicate the port to a particular type of service.

17. (original) A method as defined in claim 11, further comprising the step of:

adding a command header to data sent by the at least one component, the command header casing the port to perform a certain co-processing task to data sent by the at lest one component prior to the data being sent from the port.

18. (original) A method as defined in claim 17, wherein the co-processing task performed by the port provides for one of rate conversion and summing two or more data streams together.

19. (original) A method for providing a smart port in an interprocessor network having a component coupled to an Interprocessor Communications (IPC) stack that includes a session manager and a device layer including the smart port, the method comprising the steps of:

requesting a type of service by the component to the session manger;

negotiating the type of service between the session manger and the device layer;

determining availability of the smart port to support the type of service requested by the device layer;

granting a Service ID by the device layer if it determines that the smart port can support the requested type of service; and

forwarding the Service ID by the session manager to the component.

20. (original) A method as defined in claim 19, further comprising the step of:

sending the Service ID along with data anytime the component wants the type of service performed by the smart port.

21. (original) A method as defined in claim 20, wherein the Service ID is located in a co-processor command block.